Dynamic Web Pages Using JavaScript

Window Object

A window object represents an open window in a browser.

- A window object has properties including:
  - `innerWidth` - inner width of a window’s content area
  - `innerHeight` - inner height of a window’s content area
  - `closed` - boolean value indicating whether the window is (still) open

Methods provided by a window object include:

- `open(url, name [, features])`
  - opens a new browser window/tab
  - returns a reference to a window object
  - `url` is the URL to access in the new window; can be the empty string
  - `name` is a name given to the window for later reference
  - `features` is a string that determines various window features

The standard sequence for the creation of a new windows is not:

```javascript
var newWin = new Window (...);
newWin.document.write('<html>...<html>');
```

Instead it is:

```javascript
// new window created by using 'open' with an existing one
var newWin = window.open(,...);
newWin.document.write('<html>...<html>');
```

Example: We can write `alert()` instead of `window.alert()`.
Window Object: Dialog Boxes

- Often we only want to open a new window in order to
  - display a message
  - ask for confirmation of an action
  - request an input

- For these purposes, the window object in JavaScript provides pre-defined methods for the handling of dialog boxes (windows for simple dialogs):
  - `null alert(message_string)`
  - `bool confirm(message_string)`
  - `string prompt(message_string, default)`

Example:

```
alert("Local time: "+(new Date).toString())
```

Dialog Boxes: Example

```
<!DOCTYPE html>
<html lang="en-GB">
<head><title>Interaction example</title></head>
<body>
<script>
  do {
    string = prompt("How many items do you want to buy?");
    quantity = parseInt(string);
  } while(isNaN(quantity) || quantity <= 0);
  do {
    string = prompt("How much does an item cost?");
    price = parseFloat(string);
  } while(isNaN(price) || price < 0);
  buy = confirm("You will have to pay "+(price*quantity).toFixed(2)+"\n\ndo you want to proceed?");
  if (buy) alert("Purchase made");
</script>
</body></html>
```

User Input Validation

- A common use of JavaScript is the validation of user input in an HTML form before it is processed:
  - check that required fields have not been left empty
  - check that fields only contain allowed characters or comply to a certain grammar
  - check that values are within allowed bounds

```
<form method="post" action="process.php">
  <label>Email address: <input type="text" name="email"></label>
  <label>User name: <input type="text" name="user"></label>
  <input type="submit" name="submit" /
</form>
```

```
function validateEmail(field) {
  if (field == ") return "Valid email address";
  else return "Invalid email address"
}
```

```
function validateUser(field) {
  if (field == ") return "Valid user name";
  else return "Invalid user name"
}
```

```
function validateUserAndEmail(form) {
  fail = validateUser(form.user.value); failure = validateEmail(form.email.value);
  if (fail || failure) return true;
  else if (fail) return false
```

http://cgi.csc.liv.ac.uk/~ullrich/COMP519/examples/jsPrompt.html

http://cgi.csc.liv.ac.uk/~ullrich/COMP519/examples/jsValidate.html

http://cgi.csc.liv.ac.uk/~ullrich/COMP519/examples/jsValidate.html
Revision and Further Reading

- Read
  - Chapter 21: Introduction to JavaScript: The Browser Object
  - Chapter 22: Using JavaScript: Meet the DOM

  O’Reilly, 2018.
  E-book https://library.liv.ac.uk/record=b5647021

- Read
  - Chapter 8: The Browser Object Model
  - Chapter 9: Client Detection

  Harold Cohen Library 518.59.Z21 or
  E-book http://library.liv.ac.uk/record=b2238913